

Tipos MIME (tipos de mídia da IANA)

Um tipo de mídia (também conhecido como Extensões de Correio da Internet Multipropósito ou tipo MIME) indica a natureza e o formato de um documento, arquivo ou conjunto de bytes.

Os tipos MIME são definidos e padronizados no RFC 6838 da IETF.

A Autoridade de Números Atribuídos da Internet (IANA) é responsável por todos os tipos MIME oficiais, e você pode encontrar a lista mais atualizada e completa na sua página de Tipos de Mídia.

Estrutura de um tipo MIME

Um tipo MIME geralmente consiste em apenas duas partes: um tipo e um subtipo, separados por uma barra (/) — sem espaços em branco entre eles:

```
bash Copiar Editar
tipo/subtipo
```

O tipo representa a categoria geral na qual o tipo de dado se enquadra, como video ou text.

O subtipo identifica o tipo exato de dado do tipo especificado que o tipo MIME representa. Por exemplo, para o tipo MIME text, o subtipo pode ser plain (texto simples), html (código-fonte HTML) ou calendar (para arquivos iCalendar/.ics).

⚠ Aviso: Navegadores usam o tipo MIME, e não a extensão do arquivo, para determinar como processar uma URL, portanto é importante que os servidores web enviem o tipo MIME correto no cabeçalho Content-Type da resposta. Se isso não for configurado corretamente, os navegadores provavelmente interpretarão erroneamente o conteúdo dos arquivos, os sites não funcionarão corretamente e arquivos baixados poderão ser tratados de forma inadequada.

Cada tipo possui seu próprio conjunto de subtipos possíveis. Um tipo MIME sempre possui tanto um tipo quanto um subtipo, nunca apenas um ou outro.

Um parâmetro opcional pode ser adicionado para fornecer detalhes adicionais:

```
bash Copiar Editar
tipo/subtipo;parâmetro=valor
```

Por exemplo, para qualquer tipo MIME cujo tipo principal seja text, você pode adicionar o parâmetro opcional charset para especificar o conjunto de caracteres usado nos dados. Se nenhum charset for especificado, o padrão é ASCII (US-ASCII), a menos que seja sobrescrito pelas configurações do agente do usuário. Para especificar um arquivo de texto em UTF-8, usa-se text/plain;charset=UTF-8.

Os tipos MIME não diferenciam maiúsculas de minúsculas, mas são tradicionalmente escritos em minúsculas. Os valores de parâmetros podem diferenciar maiúsculas de minúsculas.

Tipos

Há duas classes de tipos: **discretos** e **multipartes**. Tipos discretos são tipos que representam um único arquivo ou mídia, como um único arquivo de texto ou música, ou um único vídeo. Um tipo multiparte representa um documento composto de múltiplas partes componentes, cada uma das quais pode ter seu próprio tipo MIME individual; ou, um tipo multiparte pode encapsular múltiplos arquivos sendo enviados juntos em uma transação. Por exemplo, tipos MIME multiparte são usados ao anexar múltiplos arquivos a um e-mail.

Tipos discretos

Os tipos discretos atualmente registrados na IANA são:

- **application**

Qualquer tipo de dado binário que não se enquadre explicitamente em um dos outros tipos; seja dado que será executado ou interpretado de alguma forma ou dados binários que exigem um aplicativo específico ou categoria de aplicativo para serem usados. Dados binários genéricos (ou dados binários cujo tipo real é desconhecido) são `application/octet-stream`. Outros exemplos comuns incluem `application/pdf`, `application/pkcs8` e `application/zip`.

(Veja o registro do tipo *application* na IANA)

- **audio**

Dados de áudio ou música. Exemplos incluem `audio/mpeg`, `audio/vorbis`.

(Veja o registro do tipo *audio* na IANA)

- **example**

Reservado para uso como marcador de posição em exemplos mostrando como usar tipos MIME. Estes nunca devem ser usados fora de trechos de código de exemplo e documentação. `example` também pode ser usado como subtipo; por exemplo, em um exemplo relacionado ao uso de áudio na web, o tipo MIME `audio/example` pode ser usado para indicar que o tipo é um marcador de posição e deve ser substituído por um apropriado ao usar o código no mundo real.

- **font**

Dados de fonte/tipografia. Exemplos comuns incluem `font/woff`, `font/ttf` e `font/otf`.

(Veja o registro do tipo *font* na IANA)

- **image**

Dados de imagem ou gráficos, incluindo imagens bitmap e vetoriais estáticas, bem como versões animadas de formatos de imagem estática como GIF animado ou APNG. Exemplos comuns incluem `image/jpeg`, `image/png` e `image/svg+xml`.

(Veja o registro do tipo *image* na IANA)

- **model**

Dados de modelo para um objeto ou cena 3D. Exemplos incluem `model/3mf` e `model/vrml`.

(Veja o registro do tipo *model* na IANA)

- **text**

Dados somente texto incluindo qualquer conteúdo legível por humanos, código-fonte ou dados textuais como dados formatados por valores separados por vírgula (CSV). Exemplos incluem: `text/plain`, `text/csv` e `text/html`.

(Veja o registro do tipo *text* na IANA)

- **video**

Dados ou arquivos de vídeo, como filmes MP4 (`video/mp4`).

(Veja o registro do tipo *video* na IANA)

Para documentos de texto sem um subtipo específico, deve-se usar `text/plain`.

De forma similar, para documentos binários sem um subtipo específico ou conhecido, deve-se usar `application/octet-stream`.

Tipos multipartes

Tipos multipartes indicam uma categoria de documento dividida em partes, frequentemente com diferentes tipos MIME; eles também podem ser usados — especialmente em cenários de e-mail — para representar múltiplos arquivos separados que fazem parte da mesma transação. Eles representam um documento composto.

Exceto por `multipart/form-data`, usado no método POST de formulários HTML, e `multipart/byteranges`, usado com o código de status **206 Partial Content** para enviar parte de um documento, o HTTP não trata documentos multipartes

de forma especial: a mensagem é transmitida ao navegador (que provavelmente exibirá uma janela "Salvar como" se não souber como exibir o documento).

Existem dois tipos multipartes:

- **message**
Uma mensagem que encapsula outras mensagens. Isso pode ser usado, por exemplo, para representar um e-mail que inclui uma mensagem encaminhada como parte dos seus dados, ou para permitir o envio de mensagens muito grandes em pedaços como se fossem várias mensagens. Exemplos incluem `message/rfc822` (para mensagens citadas encaminhadas ou respondidas) e `message/partial` para permitir dividir uma mensagem grande em partes menores automaticamente a serem remontadas pelo destinatário. (Veja o registro do tipo *message* na IANA)
 - **multipart**
Dados que consistem em múltiplos componentes que podem individualmente ter diferentes tipos MIME. Exemplos incluem `multipart/form-data` (para dados produzidos usando a API `FormData`) e `multipart/byteranges` (definido no RFC 7233, seção 5.4.1 e usado com a resposta HTTP 206 "Partial Content", retornada quando os dados buscados são apenas parte do conteúdo, como é entregue usando o cabeçalho `Range`). (Veja o registro do tipo *multipart* na IANA)
-

Tipos MIME importantes para desenvolvedores Web

- **application/octet-stream**
Este é o padrão para arquivos binários. Como significa arquivo binário desconhecido, os navegadores geralmente não o executam, ou mesmo perguntam se deve ser executado. Eles o tratam como se o cabeçalho `Content-Disposition` estivesse definido como `attachment`, e propõem uma janela "Salvar como".
- **text/plain**
Este é o padrão para arquivos textuais. Mesmo que realmente signifique "arquivo textual desconhecido", os navegadores presumem que podem exibi-lo.
- **text/css**
Arquivos CSS usados para estilizar uma página Web devem ser enviados com `text/css`. Se um servidor não reconhecer o sufixo `.css` para arquivos CSS, ele pode enviá-los com os tipos MIME `text/plain` ou `application/octet-stream`. Se isso ocorrer, eles não serão reconhecidos como CSS pela maioria dos navegadores e serão ignorados.
- **text/html**
Todo conteúdo HTML deve ser servido com este tipo. Tipos MIME alternativos para XHTML (como `application/xhtml+xml`) são em sua maioria inúteis hoje em dia.

Nota: `text/plain` não significa "qualquer tipo de dado textual". Se esperarem um tipo específico de dado textual, provavelmente não o considerarão compatível.

Especificamente, se baixarem um arquivo `text/plain` de um elemento `<link>` declarando um arquivo CSS, não o reconhecerão como um arquivo CSS válido se for apresentado como `text/plain`. Deve-se usar o tipo MIME CSS `text/css`.

Nota: Use `application/xml` ou `application/xhtml+xml` se quiser as regras estritas de análise do XML, seções `<![CDATA[...]]>` ou elementos que não são dos namespaces HTML/SVG/MathML.

text/javascript

De acordo com o registro de Tipos de Mídia da IANA, o RFC 9239 e a especificação HTML, o conteúdo JavaScript deve sempre ser servido usando o tipo MIME `text/javascript`.

Nenhum outro tipo MIME é considerado válido para JavaScript, e usar qualquer tipo MIME diferente de `text/javascript` pode resultar em scripts que não carregam ou não executam.

Você pode encontrar algum conteúdo JavaScript servido incorretamente com um parâmetro `charset` como parte do tipo MIME — como uma tentativa de especificar o conjunto de caracteres para o conteúdo do script. Esse parâmetro `charset` não é válido para conteúdo JavaScript, e na maioria dos casos resultará em falha no carregamento do script.

application/json

JSON (JavaScript Object Notation) é um formato padrão baseado em texto para representar dados estruturados com base na sintaxe de objetos do JavaScript. É comumente usado para transmitir dados em aplicações web.

Tipos MIME legados do JavaScript

Além do tipo MIME `text/javascript`, por razões históricas, o *MIME Sniffing Standard* (a definição de como os navegadores devem interpretar tipos MIME e descobrir o que fazer com conteúdo que não possui um válido) também permite que JavaScript seja servido usando qualquer um dos seguintes tipos MIME legados:

- `application/javascript`
- `application/ecmascript`
- `application/x-ecmascript`
- `application/x-javascript`
- `text/ecmascript`
- `text/javascript1.0`
- `text/javascript1.1`
- `text/javascript1.2`
- `text/javascript1.3`
- `text/javascript1.4`
- `text/javascript1.5`
- `text/jscript`
- `text/livescript`
- `text/x-ecmascript`
- `text/x-javascript`

Nota: Mesmo que qualquer agente do usuário possa suportar qualquer um ou todos esses, você deve usar apenas `text/javascript`. É o único tipo MIME garantido para funcionar agora e no futuro.

Tipos de imagem

Arquivos cujo tipo MIME é `image` contêm dados de imagem. O subtipo especifica qual formato específico de arquivo de imagem os dados representam.

Os seguintes tipos de imagem são usados com frequência suficiente para serem considerados seguros para uso em páginas da web:

- image/apng: Gráficos Portáteis Animados (APNG)
- image/avif: Formato de Arquivo de Imagem AV1 (AVIF)
- image/gif: Formato de Intercâmbio de Gráficos (GIF)
- image/jpeg: Imagem do Grupo de Especialistas Fotográficos Conjuntos (JPEG)
- image/png: Gráficos Portáteis de Rede (PNG)
- image/svg+xml: Gráficos Vetoriais Escaláveis (SVG)
- image/webp: Formato de Imagem Web (WEBP)

O guia de tipo e formato de arquivo de imagem fornece informações e recomendações sobre quando usar os diferentes formatos de imagem.

Tipos de áudio e vídeo

Assim como ocorre com imagens, o HTML não exige que os navegadores web suportem nenhum tipo específico de arquivo e codec para os elementos <audio> e <video>, portanto, é importante considerar seu público-alvo e a gama de navegadores (e versões desses navegadores) que eles podem estar usando ao escolher o tipo de arquivo e os codecs a serem usados para mídia.

Nosso guia de formatos de contêiner de mídia fornece uma lista dos tipos de arquivo comumente suportados por navegadores da web, incluindo informações sobre quais são seus casos de uso especiais, quaisquer desvantagens que tenham, e informações de compatibilidade, juntamente com outros detalhes.

Os guias de codec de áudio e de vídeo listam os diversos codecs que os navegadores costumam suportar, fornecendo detalhes de compatibilidade juntamente com informações técnicas como quantos canais de áudio eles suportam, que tipo de compressão é usada, e quais taxas de bits são úteis, entre outras.

O guia dos codecs usados pelo WebRTC amplia isso cobrindo especificamente os codecs suportados pelos principais navegadores, para que você possa escolher os codecs que melhor abrangem a variedade de navegadores que deseja suportar.

Quanto aos tipos MIME de arquivos de áudio ou vídeo, eles normalmente especificam o formato de contêiner (tipo de arquivo). O parâmetro opcional codecs pode ser adicionado ao tipo MIME para especificar ainda mais quais codecs usar e quais opções foram usadas para codificar a mídia, como perfil de codec, nível ou outras informações semelhantes.

Para mais informações sobre tipos de mídia comuns, consulte a página **Common MIME types**.

multipart/form-data

O tipo multipart/form-data pode ser usado ao enviar os valores de um formulário HTML preenchido do navegador para o servidor.

Como um formato de documento multiparte, ele consiste em diferentes partes, delimitadas por uma **fronteira** (uma string começando com dois hífen --). Cada parte é uma entidade própria com seus próprios cabeçalhos HTTP, Content-Disposition e Content-Type para campos de envio de arquivos.

HTML

html

Copiar

Editar

```
<form
  action="http://localhost:8000/"
  method="post"
  enctype="multipart/form-data">
  <label>Name: <input name="myTextField" value="Test" /></label>
  <label><input type="checkbox" name="myCheckBox" /> Check</label>
  <label>
    Upload file: <input type="file" name="myFile" value="test.txt" />
  </label>
  <button>Send the file</button>
</form>
```

HTTP

h

Copiar

Editar

```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:50.0) Gecko/20100101
Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Content-Type: multipart/form-data; boundary=-----
-8721656041911415653955004498
Content-Length: 465
```

Corpo da requisição:

pgsql

Copiar

Editar

```
-----8721656041911415653955004498
Content-Disposition: form-data; name="myTextField"

Test
-----8721656041911415653955004498
Content-Disposition: form-data; name="myCheckBox"

on
-----8721656041911415653955004498
Content-Disposition: form-data; name="myFile"; filename="test.txt"
Content-Type: text/plain

Simple file.
-----8721656041911415653955004498--
```

multipart/byteranges

O tipo MIME multipart/byteranges é usado para enviar respostas parciais ao navegador.

Quando o código de status **206 Partial Content** é enviado, esse tipo MIME indica que o documento é composto de várias partes, uma para cada um dos intervalos solicitados.

Como outros tipos multipart, o cabeçalho Content-Type usa uma boundary (fronteira) para separar as partes. Cada parte tem um cabeçalho Content-Type com seu tipo real e um Content-Range indicando o intervalo que representa.

HTTP

```
h HTTP/1.1 206 Partial Content
Accept-Ranges: bytes
Content-Type: multipart/byteranges; boundary=3d6b6a416f9b5
Content-Length: 385
```

Corpo da resposta:

```
php-template --3d6b6a416f9b5
Content-Type: text/html
Content-Range: bytes 100-200/1270

<meta http-equiv="Content-type" content="text/html; charset=utf-8" />

--3d6b6a416f9b5
Content-Type: text/html
Content-Range: bytes 300-400/1270

<style>
  body {
    background-color: #f0f0f2;
    margin: 0;
    padding: 0;
    font-family: "Open Sans", "Helvetica";
  }
</style>
--3d6b6a416f9b5--
```

Importância de configurar o tipo MIME corretamente

Algumas configurações de servidor podem usar o tipo MIME associado para realizar otimizações, como concatenação de arquivos, compressão ou cache. Veja [h5bp/server-configs-apache](#) para um exemplo de configuração do Apache que comprime arquivos de certos tipos MIME.

A maioria dos servidores web envia recursos não reconhecidos como o tipo MIME `application/octet-stream`. Por motivos de segurança, a maioria dos navegadores não permite definir uma ação padrão personalizada (como "Abrir no Word") para esses recursos, forçando o usuário a salvá-los em disco para usá-los.

Algumas configurações incorretas comuns de servidores:

- **Arquivos comprimidos RAR**
Neste caso, o ideal seria o tipo verdadeiro dos arquivos originais; isso é frequentemente impossível, pois arquivos .RAR podem conter vários recursos de diferentes tipos. Nesse caso, configure o servidor para enviar `application/x-rar-compressed`.
 - **Áudio e vídeo**
Somente recursos com o tipo MIME correto serão reproduzidos em elementos `<video>` ou `<audio>`. Certifique-se de especificar o tipo de mídia correto para áudio e vídeo.
 - **Tipos de arquivos proprietários**
Um tipo específico como `application/vnd.ms-powerpoint` permite que os usuários abram tais arquivos automaticamente no software de apresentação de sua escolha.
-

Sniffing MIME

Na ausência de um tipo MIME, ou em certos casos onde os navegadores acreditam que estão incorretos, os navegadores podem realizar *sniffing MIME* — adivinhando o tipo MIME correto ao examinar os bytes do recurso.

Cada navegador realiza sniffing MIME de maneira diferente e sob diferentes circunstâncias. (Por exemplo, o Safari examinará a extensão do arquivo na URL se o tipo MIME enviado for inadequado.) Existem preocupações de segurança, pois alguns tipos MIME representam conteúdo executável. Os servidores podem evitar o sniffing MIME enviando o cabeçalho X-Content-Type-Options.

Outros métodos de indicar o tipo de documento

Tipos MIME não são a única forma de indicar informações sobre o tipo de documento:

- **Sufixos de nome de arquivo**
São usados às vezes, especialmente no Microsoft Windows. Nem todos os sistemas operacionais consideram esses sufixos significativos (como Linux e macOS), e não há garantia de que estejam corretos.
- **Números mágicos (magic numbers)**
A sintaxe de diferentes formatos permite inferência do tipo de arquivo ao olhar para sua estrutura de bytes. Por exemplo, arquivos GIF começam com o valor hexadecimal 47 49 46 38 39 (GIF89), e arquivos PNG com 89 50 4E 47 (.PNG). Nem todos os tipos de arquivo possuem números mágicos, portanto isso também não é 100% confiável.